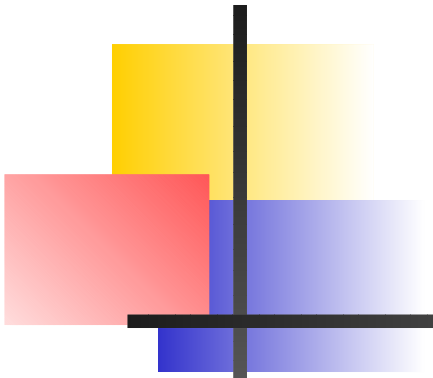


プログラミング入門 B

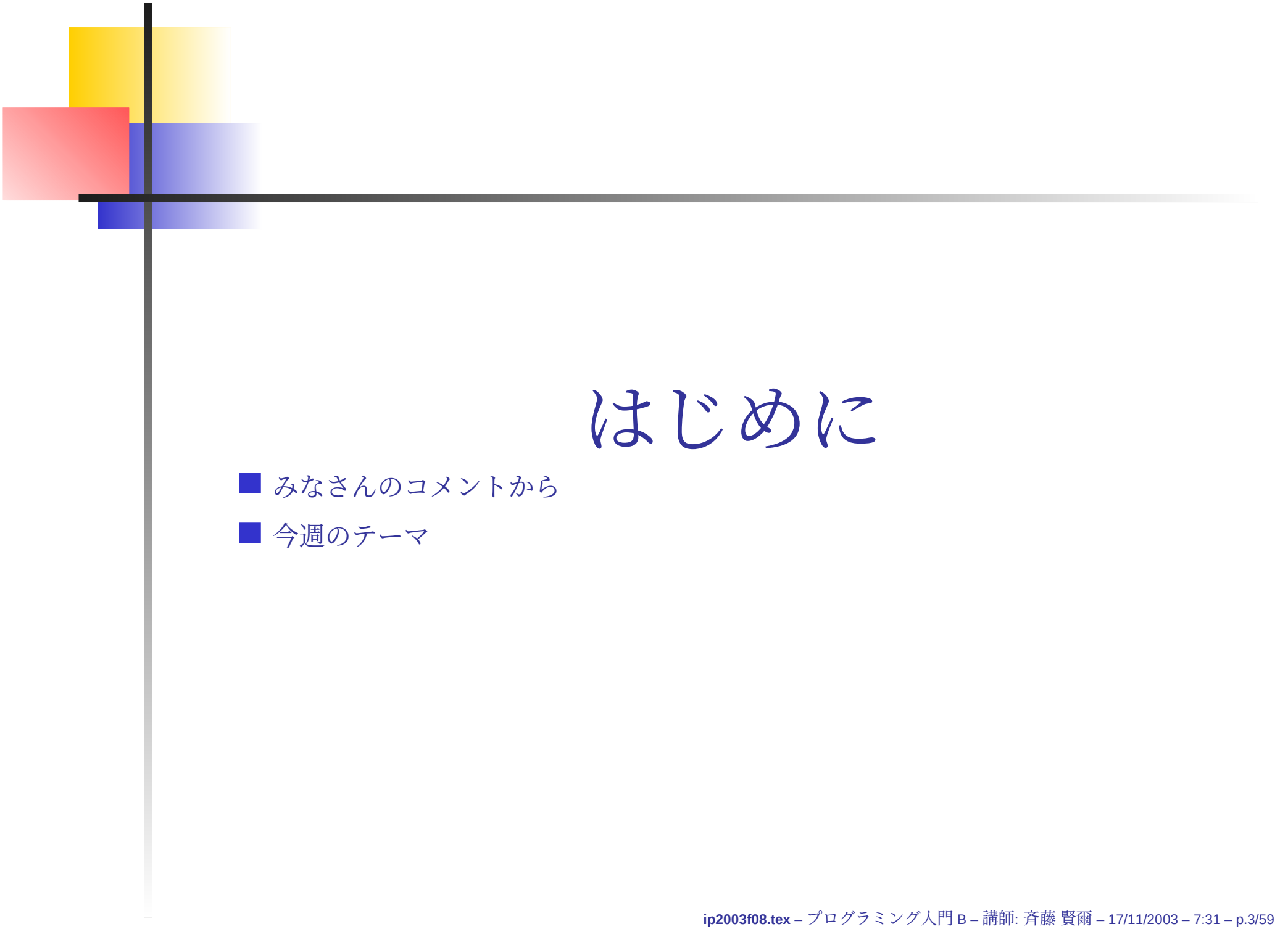
第8回「マシーンを描く」

講師: 斉藤 賢爾
ks91@sfc.wide.ad.jp
△ 棟 N206

2003 年度秋学期

今日のメニュー

- はじめに
- 課題レビュー
- 期末制作テーマの出題
- 課題の出題
- 問題には適切な表現がある
- 問題を理解するための表現の手法
- 練習問題: 自動販売機を描く
- 実習



はじめに

- みなさんのコメントから
- 今週のテーマ

みなさんのコメントから 1

- 数学的に考えるのは嫌いじゃないので、自分なりに数学的な説明を書いてみました

  (その結果、PAD でないものが提出されました)

⇒ 数学的 (=論理的) に考えるのはとてもよいことだと思います

- ただ、それを自分なりに表現すればよいのかというと、そうではなく、相手に分かってもらえる共通の方法で表現する必要があります
  - それが実は数学の役目です (数式は誰が読んでも同じ意味になります)
- そして実は、相手に分かりやすく伝えるという点では、個性を発揮しないことが大切なのです
  - とてもユニークなことを考えたとしても、それが相手に伝わるためには、相手と共通の (従ってユニークではない) 方法でそれを表現する必要があります

みなさんのコメントから 2

- PAD クイズで、サイコロは手にとるところから始まるのに、なぜノックの方ではいきなり始まるのか混乱しました

⇒ 確かにそうですね

- PAD で描いても、中身を日本語で書いている限り、曖昧さがあって、抽象するレベル (何をひとつのことと見なして考えるかということ) が上下することがあります
  - 問題が個性を発揮してしまったということですね
    (実生活では、多くの問題が個性を発揮します)
- 逆に、だからプログラミング言語などの形式的な言語で厳密に表現することが大切だというレッスンだと思っていただければ幸いです

みなさんのコメントから 3

- PAD は、初めて接する私たちにとってはとっつきにくく表現しにくいものです
- 何でこんな簡単なことが分からないのかという感じで答えを求められても困ります
  - 反復して練習することが大切なのではないでしょうか

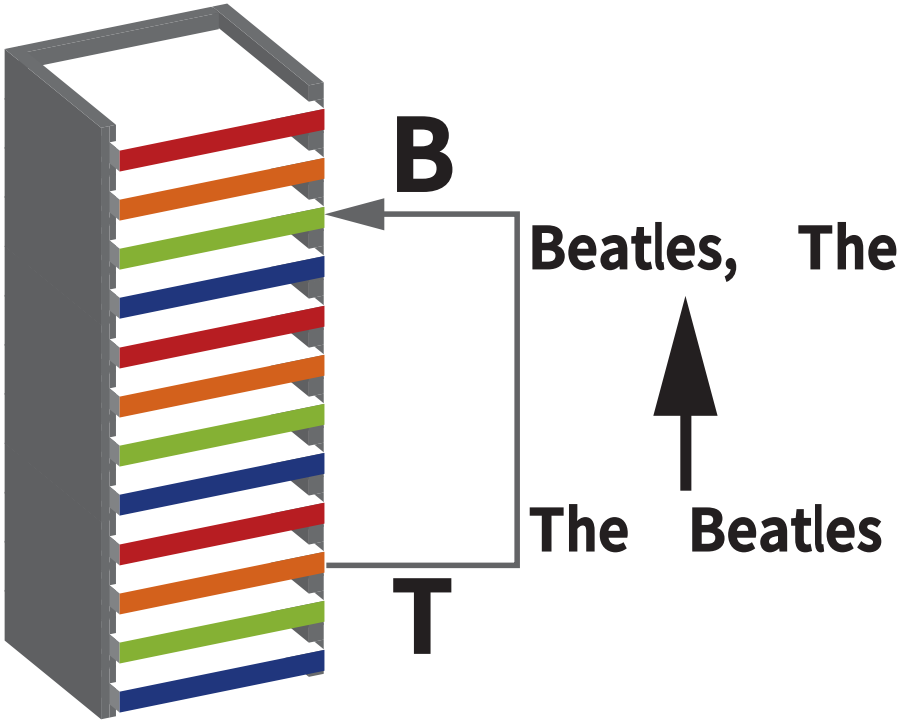
(この方自身は PAD で提出する課題に正解しています)

⇒ まさにその通りだと思います

- PAD がみなさんにとっては馴染みにくいことは知っています
- なぜ、こんな簡単なことが . . . という調子になることがありますが、それは PAD を理解しているかを問題にしているのではなく、
  - 大学生に日本語で日常的な問題の解き方を尋ねても答えられないということや
  - 他人の間違った説明やレポートの内容を無批判に採り入れてしまうという姿勢

  を問題にしているのです
- 日常的に自力で何かを解決するということの反復練習が足りないのでしょうか
  - SFC で「プログラミング入門」が必修であることには、そんな理由もあるのでしょう

ⓟ考えてみよう



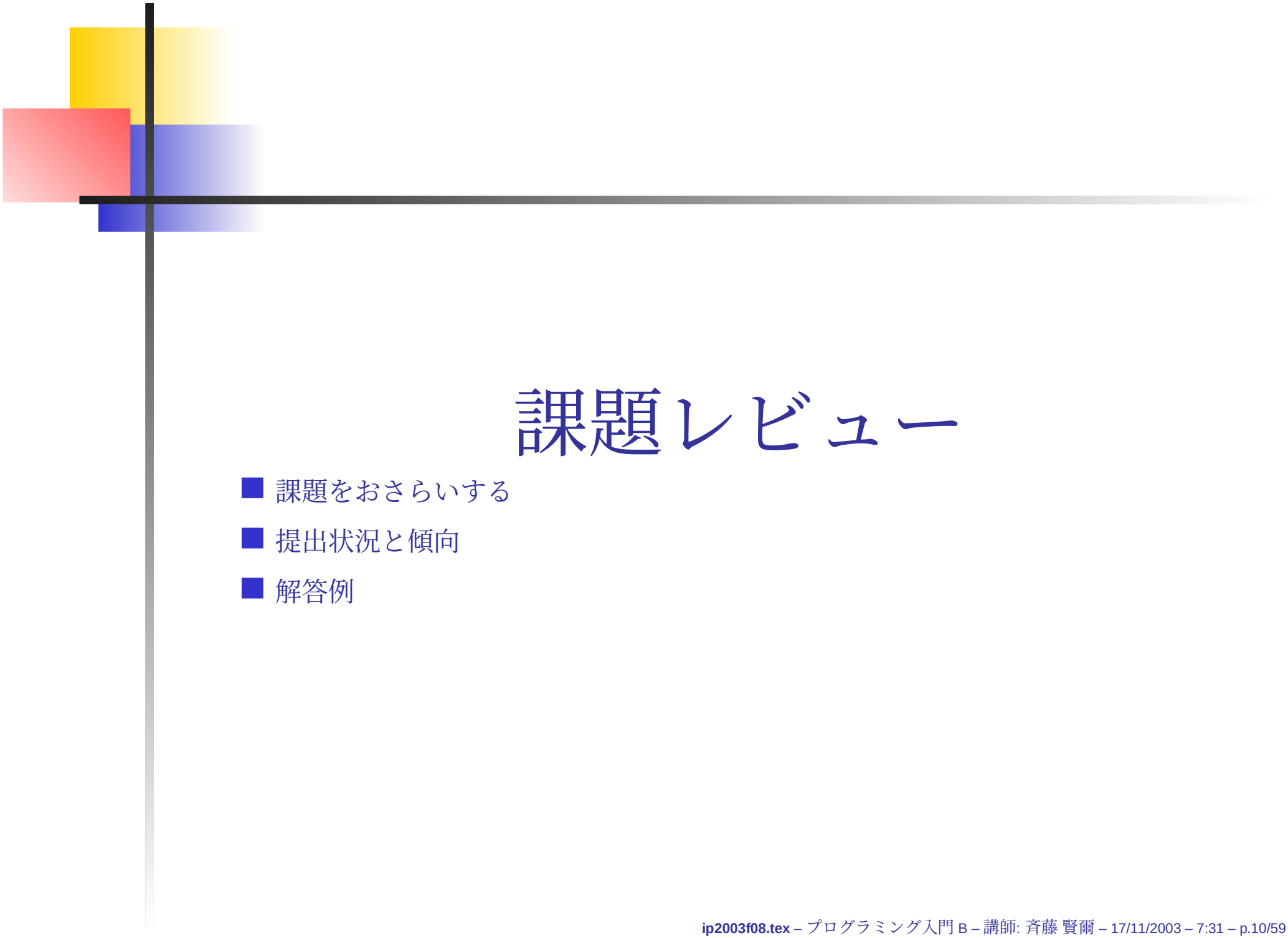
- CD ラックがあります
  - この人は几帳面なので、アーティスト名のアルファベット順に CD を並べています
- さて、The Beatles の CD を T のところに入れていたのですが、分かりにくいので B のところに入れ直すことにしました
- どのような手順で CD を並べかえますか?

今週のテーマ

- マシーンを描く
  - 身の回りの物体を、《外界からのイベントにより状態が変化する機械》と見なして表現します
    - このような機械を状態機械 (state machine) と呼びます
    - 状態機械は、ソフトウェアの設計はもちろん、日常的に私たちが使う機械 (腕時計や家電など) の取扱説明書などでも利用されている、物の理解の仕方です
      - 書画カメラで実例 (YAMAHA DX-7) を見てみましょう
  - Java 言語におけるオブジェクト (クラスやインスタンス) も、状態機械と見なすことができます
    - オブジェクトは状態 (データの集まり) を隠し持っていて
    - 外界からのイベント (メソッドの行使) により状態を変化させながら動きます
  - 今日の授業では、状態機械を図で表す方法を学びます

今日のメニュー

- はじめに
- 課題レビュー
- 期末制作テーマの出題
- 課題の出題
- 問題には適切な表現がある
- 問題を理解するための表現の手法
- 練習問題: 自動販売機を描く
- 実習



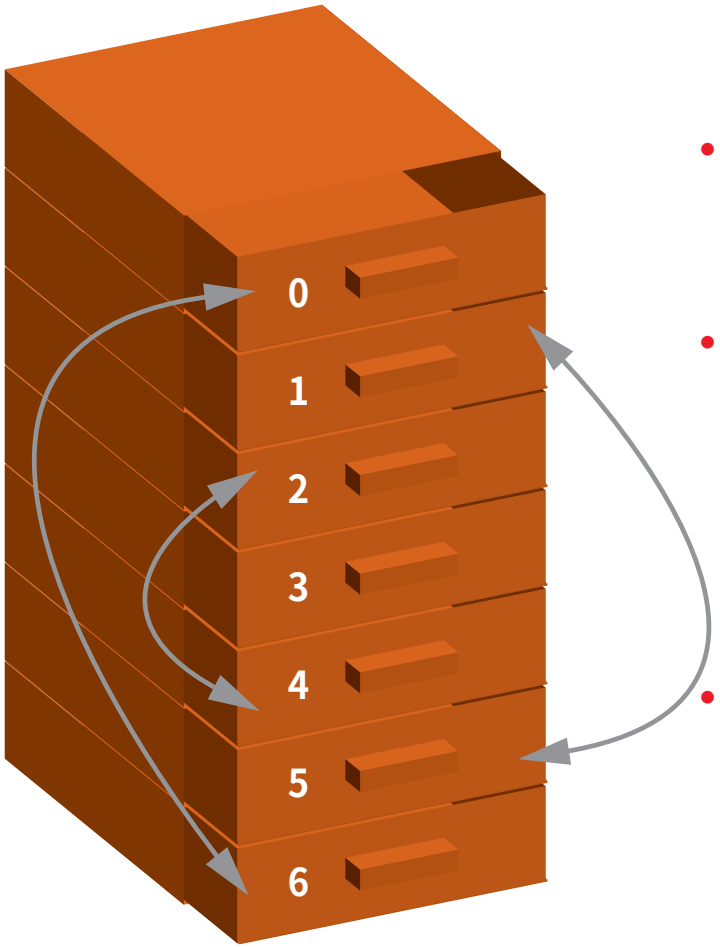
課題レビュー

- 課題をおさらいする
- 提出状況と傾向
- 解答例

課題5. ひっくり返るタンス (PAD編)

- 同じ形の引出しが縦に n 個並んでいるタンスを考えてください。
  - ただし、n は 1 以上です
  - 数学っぽくて嫌だと思った人は、○(マル)個だとおもってください
- その引出しに入っている物 (または引出しそのもの) の並びを、上下に反転させるアルゴリズムを考えて、設計図を描いてください。
  - 後述する PAD (Problem Analysis Diagram) 形式の図を描いて提出してください。
  - 手続きの名前は reverse() とします。

ⓟタンスの引出し (の中身) の並びかえ



- タンスの引き出しを上下逆に並びかえます
- 1番上の中身は　1番下に来ます
- 引き出しの数が1個以上の場合の
  アルゴリズムを考えてください

ⓟPAD とは

- Problem Analysis Diagram (問題分析図)
  - a・nal・y・sis
    1. The separation of an intellectual or substantial whole into its constituent parts for individual study. –American Heritage Dictionary
- 問題 (の解決) を、その構成要素に分解して、ひとつひとつを調べられるようにした図
  - 問題を理解すること = その問題の解決
- アルゴリズムの 3 つの制御構造を絵で表します
  - 連接
    - 例: XXX の次に YYY を行ないます
  - 選択
    - 例: XXX だったら YYY を行ない、そうでなかったら ZZZ を行ないます
  - 反復
    - 例: XXX な間、YYY を繰り返します

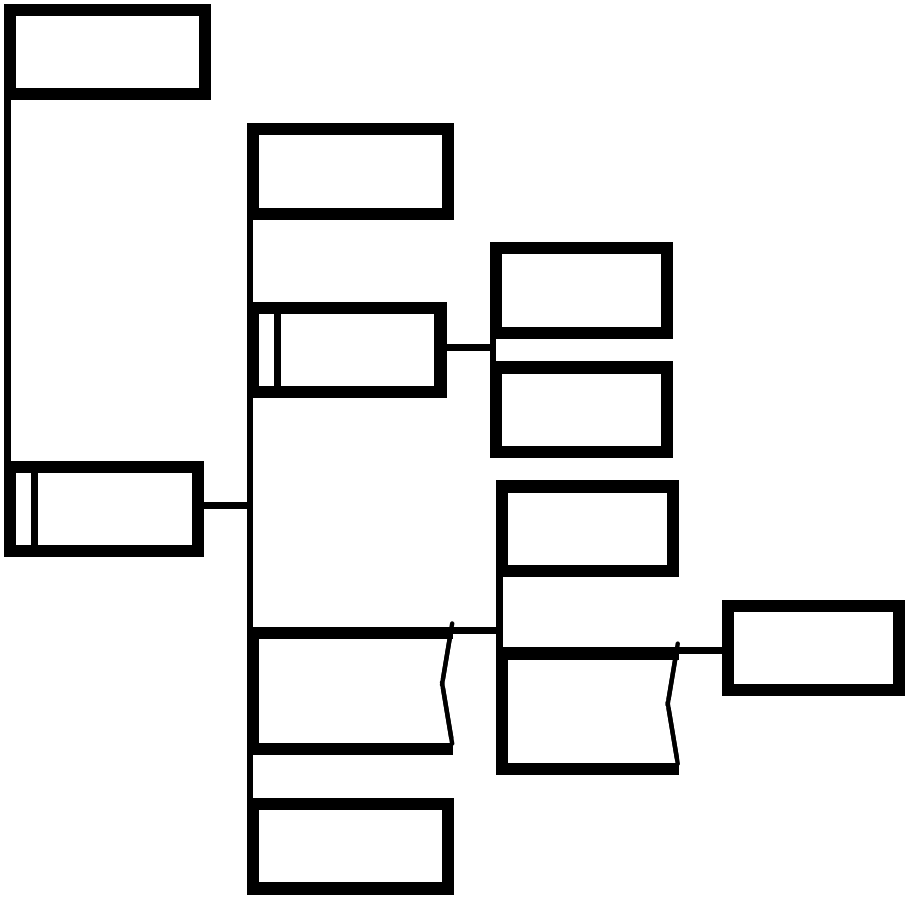
ⓟPAD の基本 ～3つの図形

意味	比喩	形
処理	箱	
選択	旗/ギザギザ	
反復	モーター	

- 直線で結び、線は折り曲げません
- 上から下に読み、ジャンプしません
- 決まった規則に従えば、どんどんつなげていけます

ⓟPAD はどんどんつなげていける

- 決まった規則に従えば、PAD はどんどんつなげていけます
  - ↓どこを切っても PAD になっています



課題6. ひっくり返るタンス (Java編)

- 授業ページからダウンロードできる ChestKeeper.java と ChestOfDrawers.java を改造し、次のように文字列の並びを反転させるプログラムを作ってください。

```
z:> java ChestKeeper foo bar baz <Enter>
baz ← 結果
bar
foo
```

  - 1 個以上の任意の個数の文字列を扱えるようにしてください。
  - ソースコード (ChestKeeper.java と ChestOfDrawers.java の両方) と実行結果 (上の例以外) を提出してください。

ⓙ課題6. のヒント

```
for (int i = 0, j = 10; 条件; i = i + 1, j = j - 1) {
        繰り返しの中身
}
```

- for 文でふたつの変数を扱う書き方の例です
  - 意味
    - i は 0 から、j は 10 から始まり、
    - i を 1 ずつ増やし、j を 1 ずつ減らしながら、
    - 条件が満たされる間、繰り返しの中身を行ないます
- ‘,’ で区切ることで、複数の文を初期化式や増減式に書けます

ⓙ1からnまで足し算するプログラム

```
public class Sum {
    public static void main(String[] aosArguments) {
        final int I_MAX = Integer.parseInt(aosArguments[0]);
        int iSum = 0;
        for (int i = 1; i <= I_MAX; i = i + 1) {
            iSum = iSum + i;
        }
        System.out.println(iSum);
    }
}
```

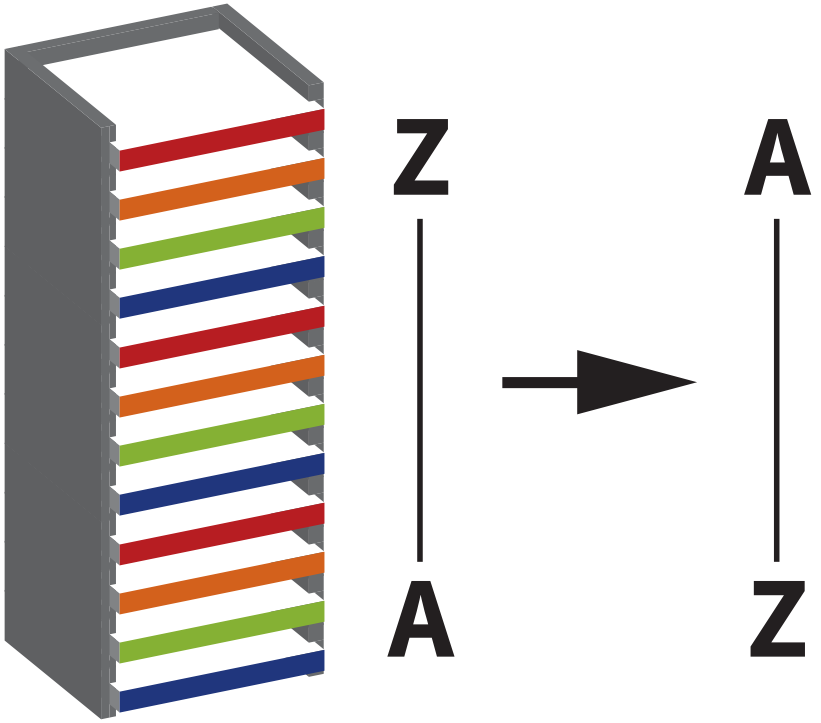
課題7. エラーに負けないプログラム

- 前ページの、1 から n まで足し算するプログラムを改良して、入力の誤りをチェックするようにしてください。
  - 次の誤りをチェックしてください:
    - プログラムの引数がない。
    - プログラムの引数が多過ぎる。
    - プログラムの引数が整数でない。
    - 0 以下の整数が指定された。
    - 65536 以上の整数が指定された。
  - 入力に誤りがある場合は、どんな誤りかを説明し、正しい使い方を表示してプログラムが終了するようにしてください。
  - ソースコードと実行結果を提出してください。

提出状況と傾向

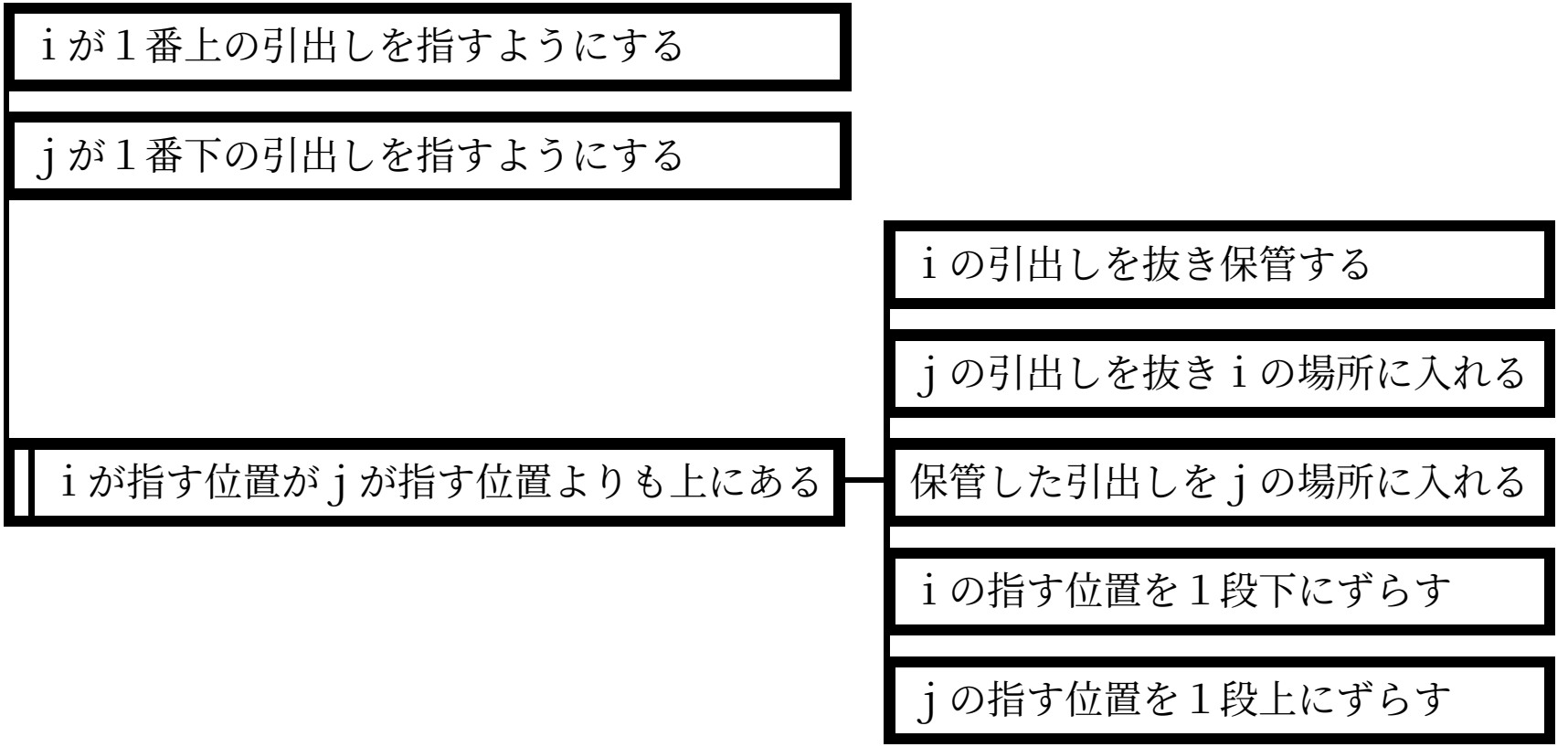
- 78 人中 課題 5: 24 人、課題 6: 24 人、課題 7: 8 人 提出 (11/17 現在)
- 傾向と所感
  - 課題 5, 6
    - 平たく書くと、どういう意味? と聞きたくなるような、必要以上に数学的な表現が目立ちました
      - 例: (n - i - 1) - i >= 0
    - $i < \frac{n}{2}$ 派と $i < j$ 派がいますが、どちらも正答できます
      - どちらが分かりやすいでしょうか
  - 課題 7
    - みなさんよくできています
    - 同じメッセージを書く労力を省いたり、エラー処理だけがまとまって先に終わるようにするといった工夫ができるともっとよかったでしょう
    - とても大きな整数が指定されたときのエラーメッセージを正しくするには、発想を転換するとよいかも知れません

ⓟ考えてみよう 2



- CD ラックがあります
  - この人は几帳面なので、アーティスト名のアルファベット順に CD を並べています
- ところが、同時におっちょこちょいなこの人は、下から順にアルファベット順に並べてしまったことに気づきました
  - そして上から順になるように並べ直そうと決心しました
- どのような手順で CD を並べかえますか?

課題5. 解答例



- 引出しが 1 つしかなくても正しく反転できます
- 引出しの数が奇数でも偶数でも大丈夫です

課題6. 解答例

(ChestKeepr.java は省略)

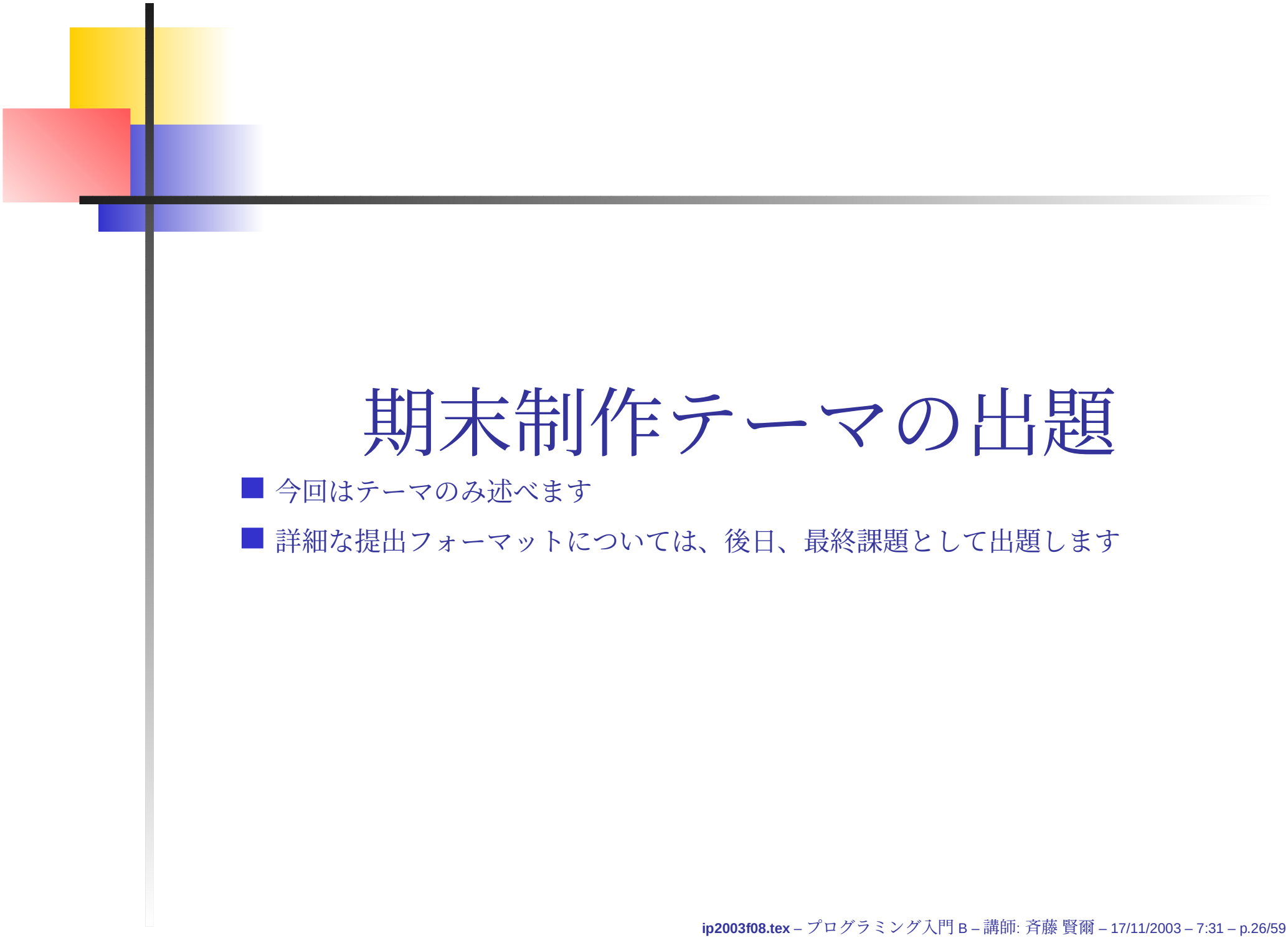
```
public void reverse() {
    //
    // i はタンスを上から下へ、j は下から上へ辿ります。
    // i が示す引出しの位置が、j が示す位置より上にあるかぎり
    // ふたつの引出しを交換することで反転を行ないます。
    //
    for (int i = 0, j = _aosThings.length -1; i < j; i = i + 1, j = j - 1) {
        String s = _aosThings[i];
        _aosThings[i] = _aosThings[j];
        _aosThings[j] = s;
    }
}
```

課題7. 解答例

- 授業ページからダウンロードしてください
  - 最初にエラーや例外の処理をまとめて済ませてから、本来やりたいことを記述するという書き方になっています
    - 経田 武志さんのレポートを参考にしました
      - 鈴木 麟太郎さん、川崎 由雅利さんも同様の書き方になっています
    - 課題７を提出できた人は、みなさんよく頑張ったと思います

今日のメニュー

- はじめに
- 課題レビュー
- 期末制作テーマの出題
- 課題の出題
- 問題には適切な表現がある
- 問題を理解するための表現の手法
- 練習問題: 自動販売機を描く
- 実習



期末制作テーマの出題

- 今回はテーマのみ述べます
- 詳細な提出フォーマットについては、後日、最終課題として出題します

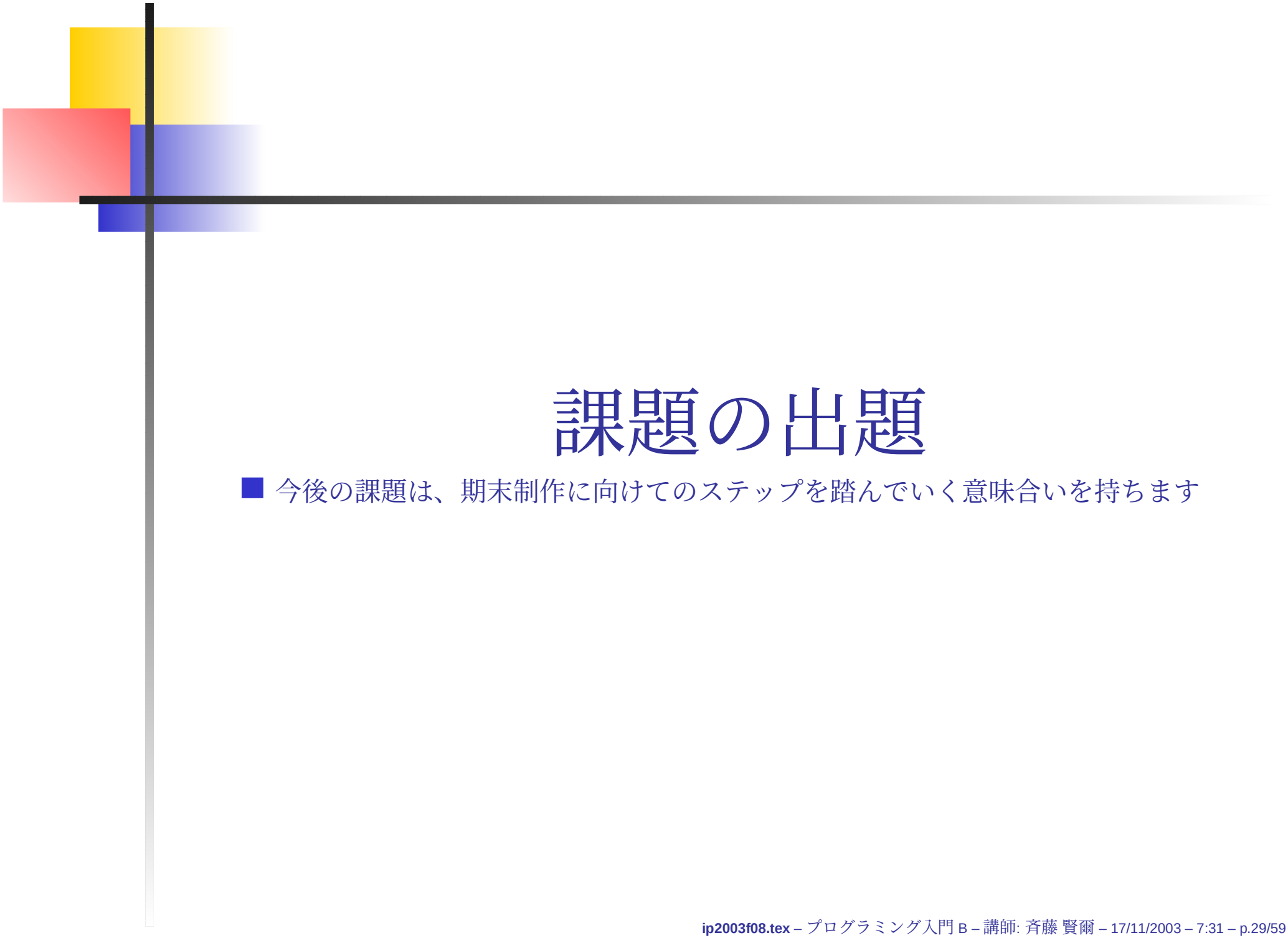
期末制作. プログラムによる物体の表現

- みなさんの身の回りに実在する物体のうち、環境との相互作用を持つものをひとつ選び、1,000 行未満程度の Java 言語のプログラムでその動作を真似てください。
  - 物体とは
    - 長さ・幅・高さの 3 次元において空間を充たしていて、知覚の対象となりうる物質。もの。
  - 環境とは
    - 周囲をとりまき、物体と相互作用を及ぼし合うものとして見た外界。
  - 相互作用とは
    - 互いに働きかけること。

  定義はすべて広辞苑第五版に依りました。
- グラフィカルなプログラムである必要はありません。
- アプリケーションとして制作してください。
  - main() メソッドを有するプログラムがアプリケーションです。
- 各自、ひとりで制作してください。

今日のメニュー

- はじめに
- 課題レビュー
- 期末制作テーマの出題
- 課題の出題
- 問題には適切な表現がある
- 問題を理解するための表現の手法
- 練習問題: 自動販売機を描く
- 実習



課題の出題

- 今後の課題は、期末制作に向けてのステップを踏んでいく意味合いを持ちます

課題8.
マシーンを描く

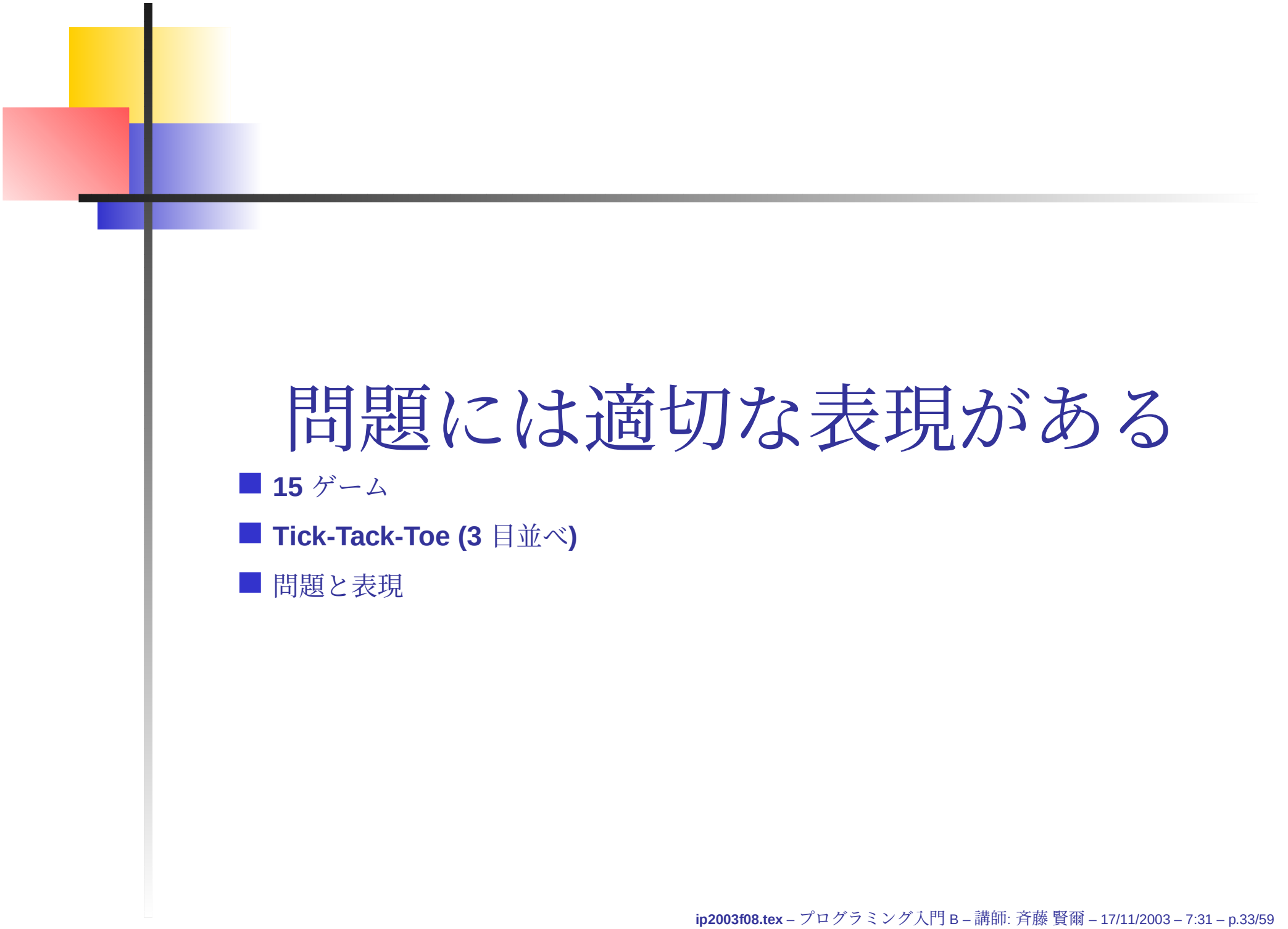
- 期末制作で真似てみたい物体を仮に決めてください。
  - 最終決定ではなく、後で変更できます。
  - まずは気楽に考えてみましょう。
- その物体は環境との相互作用によってどんな振舞いを見せるのか、図を描いて表現してください。
  - 後述する状態遷移図の形式で描いてください。
    - 物体の生成・消滅のイベントは必ず入れてください。
  - 人間が使う物体なら、その物体にとって人間は環境となります。
  - 時間の経過が外界からのイベントと捉えられる場合もあります。

提出方法

- SOI のレポートシステムを用いて提出してください
  - http://www.soi.wide.ad.jp/report/
  - レポートを画像ファイルとしてアップロードしてください
  - レポートには、真似する物体を表すタイトルをつけてください
- 〆切
  - 2003/11/30 23:59 JST

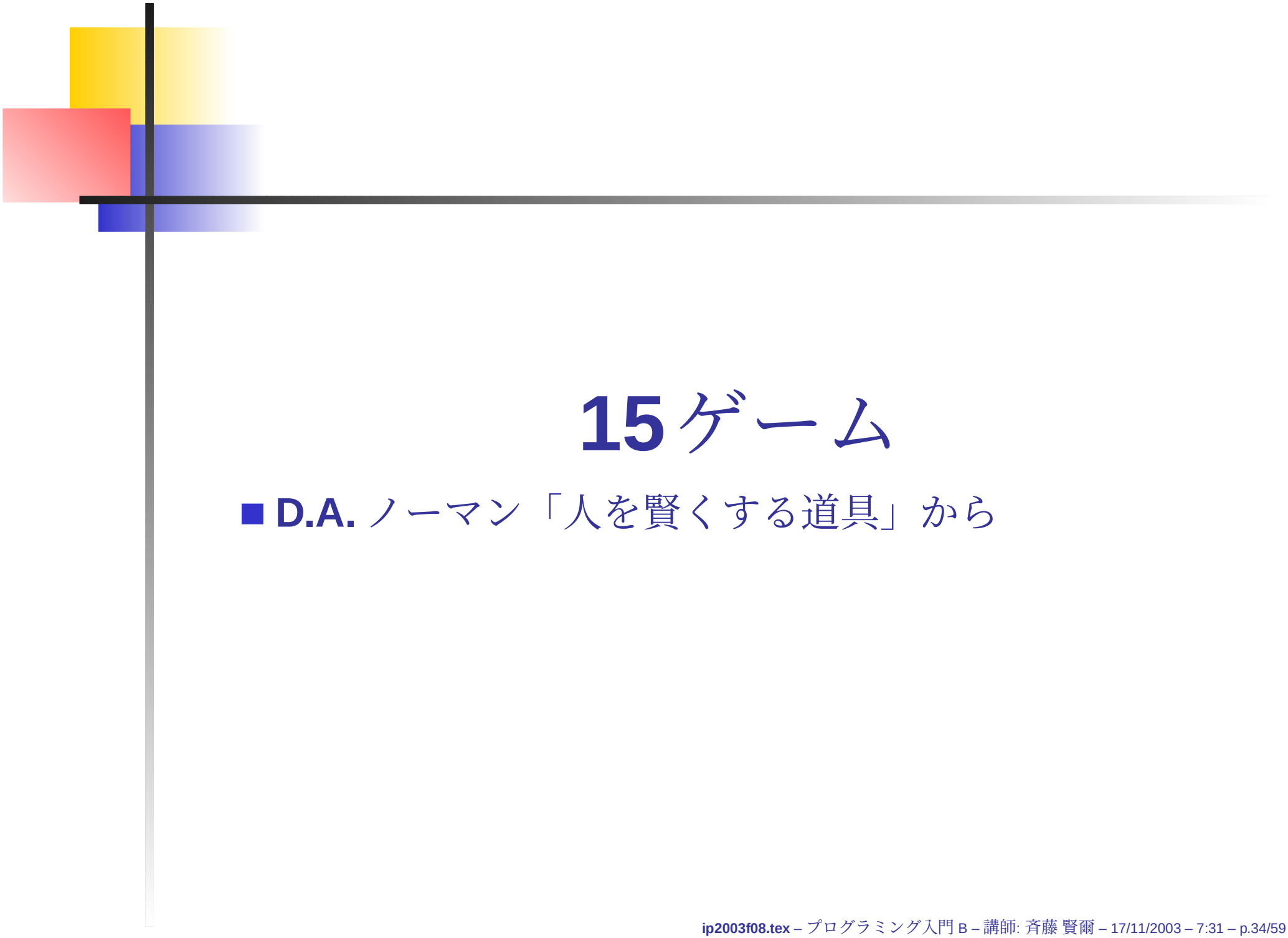
今日のメニュー

- はじめに
- 課題レビュー
- 期末制作テーマの出題
- 課題の出題
- 問題には適切な表現がある
- 問題を理解するための表現の手法
- 練習問題: 自動販売機を描く
- 実習



問題には適切な表現がある

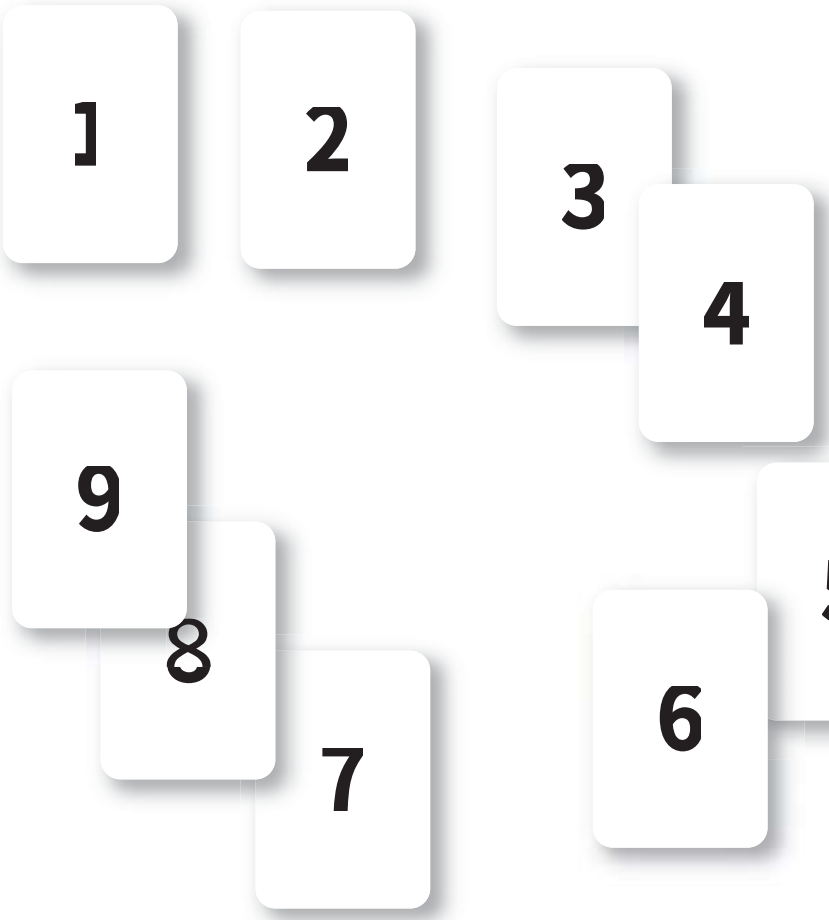
- **15** ゲーム
- **Tick-Tack-Toe (3** 目並べ**)**
- 問題と表現



15ゲーム

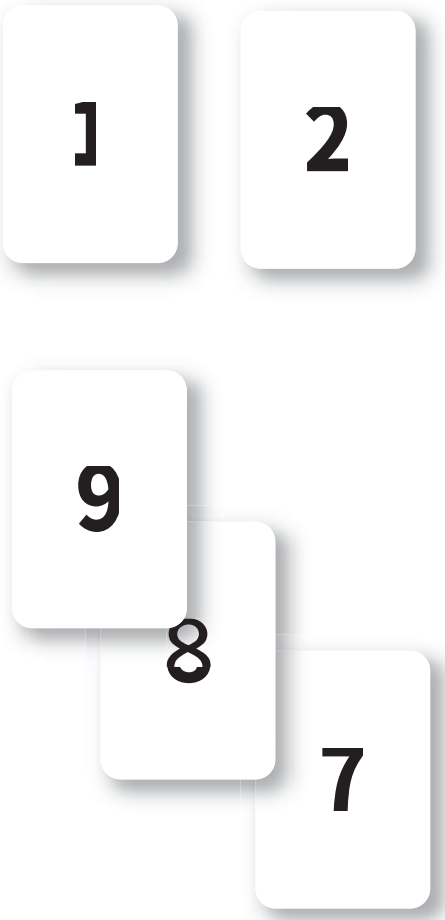
- **D.A.** ノーマン「人を賢くする道具」から

15ゲーム ～ ルール

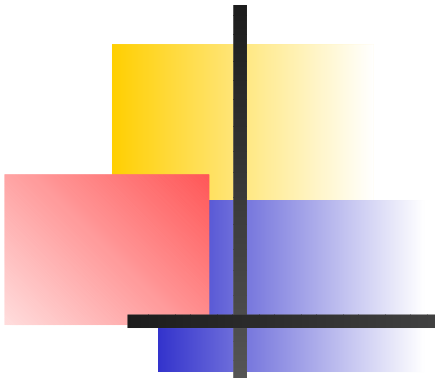


- 1 から 9 までの数字が書かれた 9 枚のカードを用います
- 2 人でプレイします
- 交互に 1 枚ずつカードを選びます
- 一度選ばれたカードはもう選べません
- 選んだカードのうち、3 枚の合計が先に 15 になった方の勝ちです

15ゲーム ～ やってみよう



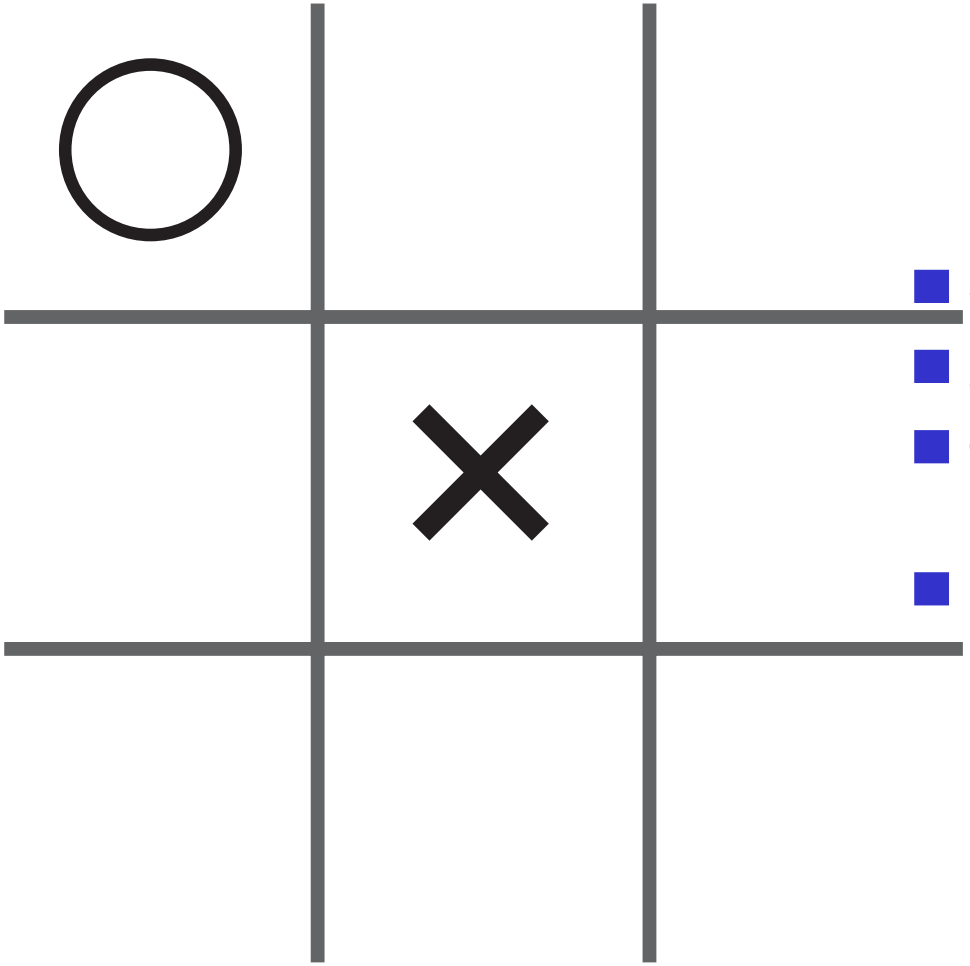
- A vs. B
  - A-8
  - B-2
  - A-4
  - B-3
  - A-5
- さぁ、B の次の手は?



Tick-Tack-Toe (3目並べ)

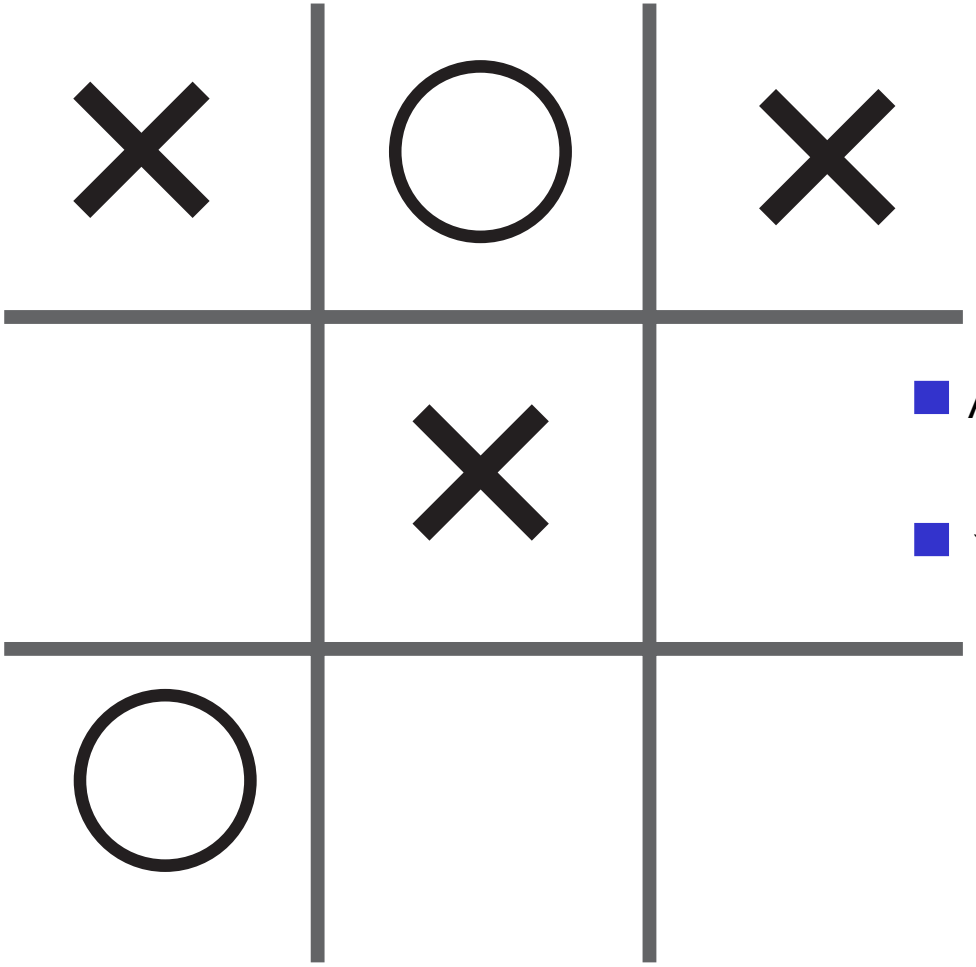
- 古典的なゲーム

Tick-Tack-Toe ～ ルール



- 井型を描きます
- 2 人でプレイします
- 9 つのスペースの中のどれかに、交互に ○/× を描きます
- 自分の描いた記号が先に 3 つ並んだ方の勝ちです

Tick-Tack-Toe
～やってみよう

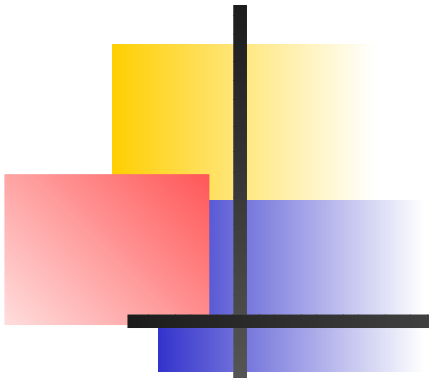


- A(×) vs. B(○)
- さぁ、B の次の手は?

ⓟ問題と表現

問題を解くということは

⇓



ⓟ問題と表現

問題を解くということは

⇓

解が自然に浮かび上がるように問題の表現を作ることに他ならない

ⓟ問題と表現

問題を解くということは

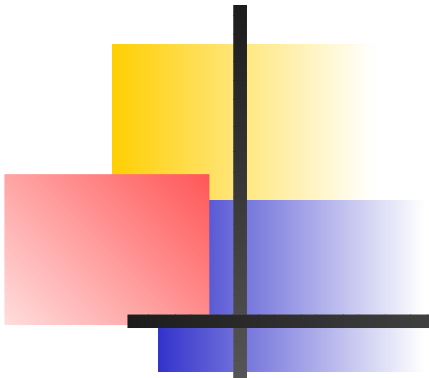
⇓

解が自然に浮かび上がるように問題の表現を作ることに他ならない

- 例えば、1 + 1 = を、ある難しい問題だと見なして、その表現を考えてみましょう
  - ある数 n の次の数を $s(n)$ で表すとします
    - $n+1$ は $s(n)$ で表すことができます
  - 1 は $s(0)$ で表すことができます
    - 0 の次の数だからです
  - 1 + 1 は $s(s(0))$ で表すことができます
    - $s(1)$ だからです
  - $s(s(0))$ は 0 の次の次の数なので、2 のことです
    - 足し算を $s(0)$ を使って表すと、そのまま答えになります

今日のメニュー

- はじめに
- 課題レビュー
- 期末制作テーマの出題
- 課題の出題
- 問題には適切な表現がある
- 問題を理解するための表現の手法
- 練習問題: 自動販売機を描く
- 実習



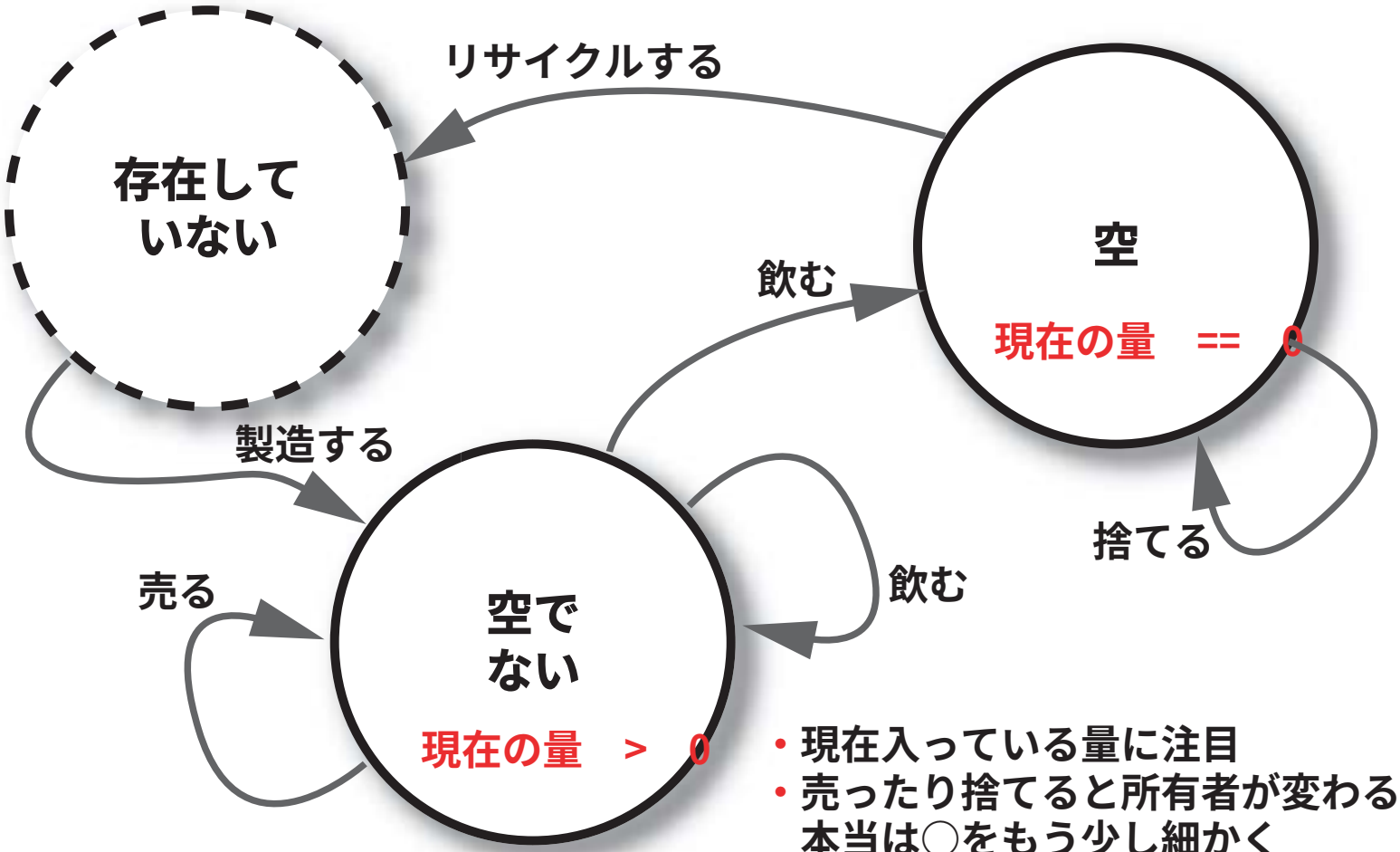
問題を理解するための表現の手法

- 状態遷移図で表現し、考える

ⓟ状態遷移図とは

- 物の一生を絵で描きます
  - 物の状態がどう変化していくかに注目します
    - ○の中にその状態を表す言葉を書いて、状態を表します
    - 実は形は何でもよいのですが、□だと PAD と混乱するかも知れないので、この授業では状態を○で表すことにします
  - どんなイベントでどのように状態が変化するかを考えます
    - Ⓐ→Ⓑ で状態 A から状態 B への遷移を表します
    - 矢印の上にイベントの説明を書きます
      - ← ホワイトボードに YAMAHA DX-7 の例を描きます
- 物について理解する上で役立つツールです
  - 「オブジェクト指向」という、プログラミング上の考え方は、人間が物について理解する能力に根ざしています
  - 「オブジェクト指向」に基づいている Java 言語でプログラムを作る上でも、状態遷移図は役に立つと考えられます

ⓟペットボトルの状態遷移図



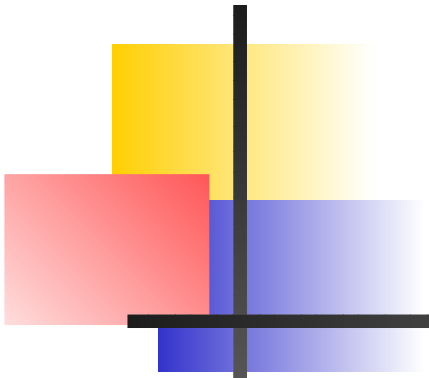
- 現在入っている量に注目
- 売ったり捨てると所有者が変わるので、本当は○をもう少し細かく分けなければならない

ⓟ状態遷移図の考え方

- 描き方
  - 物 (オブジェクト) に注目
  - 物に何が起こるかな?
  - それによって、物の状態はどう変わるかな?
- 使い方 (来週以降に詳細を説明します)
  - 物をクラスにします
  - 物が体験することをメソッドにします
  - 状態をフィールドで表して、隠蔽します
  - 構築子でフィールドを初期化します
- PAD で表現すると手続きのコードにそのまま翻訳できるように、状態遷移図で表現するとクラスとインスタンスを用いるコードにそのまま翻訳できます
  - クラス全体を表すときには状態遷移図、メソッドの中身を考えるときは PAD、というように使い分けることで設計していけます

今日のメニュー

- はじめに
- 課題レビュー
- 期末制作テーマの出題
- 課題の出題
- 問題には適切な表現がある
- 問題を理解するための表現の手法
- 練習問題: 自動販売機を描く
- 実習



練習問題: 自動販売機を描く

- 期末制作のテーマに合致する例として、簡単な自動販売機について考えてみましょう

ⓟ自動販売機はテーマに合致する

- 自動販売機は物体です
- 人間という環境との相互作用があります
  - 人間という外界が、自動販売機に対して発生させるイベントにはどんなものがありますか?
  - 自動販売機からの反応にはどんなものがありますか?

ⓙ簡単な自動販売機の例

- vending.jar
  - 授業ページからダウンロードしよう
  - 実行してみよう
    - 方法 1: vending.jar のアイコンをダブルクリック
    - 方法 2:

```
z:> java -jar vending.jar <Enter>
```

- どんな点が簡単か
  - お金も商品も無尽蔵であることが前提
  - 商品の価格は 1 種類
  - おつりは自動的には出てこない

ⓟ自動販売機を描く

- 練習として、vending.jar プログラムの状態遷移図を描いてください
- 考え方
  - どんな状態がありますか?
    - 状態が異なると、自動販売機に対してできることが異なってきます
    - 自動販売機が持つデータがどのような値を持つかで状態は異なってきます
  - どんな操作ができますか?
    - ボタンで操作するプログラムなので、どんなボタンがあるかを考えましょう
  - 操作によって、状態はどう変化しますか?

ⓟ自動販売機を描く ～考えるヒント

- どんな状態がありますか?
  - 状態が異なると、自動販売機に対してできることが異なってきます
    - ボタンがグレイアウトすると、そのボタンを押す操作はできなくなるので、その契機に注目します
  - 自動販売機が持つデータがどのような値を持つかで状態は異なってきます
    - 商品は無尽蔵なので、変化するデータは投入金額だけです
    - 投入金額が少しでもあると、おつり/取り消しボタンが押せるようになります
    - 投入金額が 120 円以上になると、商品のボタンが押せるようになります
- どんな操作ができますか?
  - ボタンで操作するプログラムなので、どんなボタンがあるかを考えましょう
    - コイン、おつり/取り消し、商品のボタンがあります
    - ウィンドウのクローズボタンも押すことができます
    - ダイアログのボタンは考えてもよいですが、無視して構いません
- 操作によって、状態はどう変化しますか?
  - 状態が変化しない操作もあります
    - 例: 120 円以上になってもコインを投入し続ける

ⓟ自動販売機を描く ～描けたらすること

- 確かめましょう
  - プログラムを作っていく過程には、それぞれ必ず plan(計画) -do(実施) -see(評価) があります
    - 何かしようと考えて、実際にやったら、必ず確かめます
  - vending.jar のプログラムを起動してから、終了するまで、どんな使い方をしても、自分が描いた状態遷移図で表されているか、確かめてください
    - 状態遷移図は、自分の都合で勝手に動くような環境を相手にするプログラム (人間相手のプログラムはすべてそうです) を記述するのに適しています

今日のメニュー

- はじめに
- 課題レビュー
- 期末制作テーマの出題
- 課題の出題
- 問題には適切な表現がある
- 問題を理解するための表現の手法
- 練習問題: 自動販売機を描く
- 実習

実習

- 残り時間を使って課題に挑戦しましょう